UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/074,092	02/12/2002	Darpan Dinker	5681-05700	6704

7590 12/06/2006

Robert C. Kowert
Conley, Rose, & Tayon, P.C.
P.O. Box 398
Austin, TX 78767

EXAMINER	
DUONG, OANH L	

ART UNIT	PAPER NUMBER
2155	

DATE MAILED: 12/06/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/074,092
Filing Date: February 12, 2002
Appellant(s): DINKER ET AL.

Robert C. Kowert
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed September 06, 2006 appealing from the Office action mailed March 21, 2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,330,689	JIN et al.	12-2001
6,097,380	CRITES et al.	08-2000

Applicant Admitted Prior Art, pp.1-4

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

1. Claims 1-9, 13-24 and 28-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Applicant Admitted Prior Art (AAPA) in view of Jin et al. (Jin) (US 6,330,689 B1).

Regarding claim 1, AAPA teaches a plurality of nodes coupled together, wherein each one of the plurality of nodes is coupled to at least one other one of the plurality of nodes for communicating data between the nodes (Figs. 1 and 2), and

in-process node (Fig. 2) comprising:

an in-process client (client 201, Fig. 2); and

a distributed data manager (Distributed Data Manager 211, Fig. 2),

wherein the in-process client and the distributed data manager for the in-process node are configured to execute within the same computer process on the in-process node (*i.e., in the in-process configuration data may be communicated between* ***a distributed***

data manager and a client sharing the same process space, *page 4 paragraph [0012])*, and wherein the distributed data manager for the in-process node is configured to communicate data with the in-process client in a non-serialized format *(i.e., in the in-process configuration data may be communicated between a distributed data manager and a client* ***without the additional computation requirement for serialization/deserialization****, page 4 paragraph [0012])* and communicate data with other ones of the plurality of nodes in a serialized format *(i.e., serialization and de-serialization take place across process boundaries, page 4 paragraph [0012]);* and

out-of-process node (Fig. 1) comprising an out-of-process client (i.e., client 101, Fig. 1), wherein the out-of-process client is configured to execute within a different process than any distributed data manager *(i.e., the distributed data manager and the client execute as different processes, page 1 paragraph [0004]),* and wherein the out-of-process client is configured to communicate data with other processes or other ones of the nodes in a serialized format *(i.e., to transmit data across boundaries, data is serialized before transmission, pages 2-3 paragraph [0007]).*

AAPA does not explicitly teach wherein the plurality of nodes in the system comprises at least one in-process node and at least one process node.

Jin teaches a server architecture wherein application can be run either in-process or out-of-process with the server program (see abstract). Jin teaches a data system comprising wherein the plurality of nodes in the system comprises at least one in-process node and at least one process node (i.e., the architecture 60 offers to run both in-process and out-of-process applications, Fig. 4, col. 6 lines 42-57).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of AAPA to run both in-process and out-of process nodes/applications in the same system as taught by Jin. One would be motivated to do so to provide higher performance at the risk of crashing the system and afford high reliability (**Jin**, col. 4 lines 55-62).

Regarding claim 2, AAPA teaches the system as recited in claim 1, wherein the in-process client is configured to request data from the distributed data manager for the in-process node, and wherein, in response to the client's request, the distributed data manager for the in-process node is configured to return a reference to an object for the data without serializing the data (page 4 lines 15-18).

Regarding claim 3, AAPA teaches the system as recited in claim 2, wherein the distributed data manager for the in-process node is configured to receive a request for data from another node, serialize the requested data and send the serialized data to the requesting node (page 4 lines 5-15).

Regarding claim 4, AAPA teaches the system as recited in claim 1, wherein the distributed data manager for the in-process node is configured to receive serialized data from another node, de-serialize the data and store the de-serialized data as an object (page 4 lines 5-15).

Regarding claim 5, AAPA teaches the system as recited in claim 1, wherein said in-process client is configured to send an object reference for the data to the distributed data manager for the in-process node to store data in the in-process node (page 3 lines 30-31).

Regarding claim 6, AAPA teaches the system as recited in claim 1, wherein all data store operations performed by the distributed data manager in the in-process node store data in a non-serialized object format in a data store of the in-process node (page 4 lines 5-18).

Regarding claim 7, AAPA teaches the system as recited in claim 1, further comprising a distributed data manager for the out-of-process node, wherein the distributed data manager for the out-of-process node is configured to communicate data with other processes or other ones of the plurality of nodes in a serialized format (page 2 line 30-page 3 line 3).

Regarding claim 8, AAPA teaches the system as recited in claim 7, wherein the out-of-process client is configured to execute in a separate process and communicate data with the distributed data manager for the out-of-process node in a serialized format (Fig. 1, page 1 line 30-page 2 line 2).

Regarding claim 9, AAPA teaches the system as recited in claim 7, wherein the out-of-process client is configured to request data from the distributed data manager for the out-of-process node, and wherein, in response to the client's request, the distributed data manager for the out-of-process node is configured to send the requested data to the out-of-process client in a serialized format (page 2 line 30-page 3 line 3).

Regarding claim 13, AAPA teaches the system as recited in claim 7, wherein the distributed data manager for the out-of-process node is configured to replicate data to one or more other ones of the plurality of nodes (page 2 lines 16-17).

Regarding claim 14, AAPA teaches the system as recited in claim 7

AAPA does not explicitly teach the distributed data manager for the out-of-process node is comprised within an application server, and wherein the out-of-process client is a web server coupled to the application server.

Jin teaches the distributed data manager for the out-of-process node is comprised within an application server, and wherein the out-of-process client is a web server coupled to the application server (col. 6 lines 30-35).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify AAPA to include the distributed data manager for the out-of-process node is comprised within an application server, and wherein the out-of-process client is a web server coupled to the application server as in Jin. One would be motivated to do so to accommodate dynamic content sessions in which the server dynamically generates and server a response that is tailored to client (Jin, col. 5 lines 62-65).

Regarding claim 15, AAPA teaches the system as recited in claim 1, wherein the distributed data manager for the in-process node is configured to replicate data stored in the in-process node to one or more other ones of the plurality of nodes (page 3 line 30-page 4 line 2).

Regarding claim 16, AAPA teaches the system as recited in claim 7

AAPA does not explicitly teach the distributed data manager for the in-process node is comprised within an application server, and wherein the in-process client is a web server coupled to the application server.

Jin teaches the distributed data manager for the in-process node is comprised within an application server, and wherein the in-process client is a web server coupled to the application server (col. 6 lines 30-35).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify AAPA to the distributed data manager for the in-process node is comprised within an application server, and wherein the in-process client is a web

server coupled to the application server as in Jin. One would be motivated to do so to accommodate dynamic content sessions in which the server dynamically generates and server a response that is tailored to client (Jin, col. 5 lines 62-65).

Regarding claim 17, AAPA teaches a method, comprising:

an in-process client requesting data from a distributed data manager for an in-process node of a distributed data system (page 4 lines 4-5), wherein the in-process client and the distributed data manager for the in-process node execute within the same process on the in-process node (Fig. 2, page3 lines 15-16);

if the requested data is present in a data store managed by the distributed data manager for the in-process node:

the distributed data manager for the in-process node returning the requested data to the in-process client as an object without serializing the data (page 4 lines 3 and lines 15-18);

if the requested data is not present in the data store managed by the distributed data manager for the in-process node:

the distributed data manager for the in-process node retrieving the requested data in a serialized format from another node of the distributed data system (page 4 lines 3-8 and lines 14-15);

the distributed data manager for the in-process node de-serializing the data retrieved from another node into an object (page 4 lines 12-15); and

the distributed data manager for the in-process node returning the requested data to the in-process client as the de-serialized object (page 4 linens 8-10);

an out-of-process client requesting data from a node (Fig.2, page 2 lines 20-21); and

the out-of-process client receiving the requested data in a serialized format (page 2 line 30-page 3 line 3).

AAPA does not explicitly teach in-process node and out-of-process node are in the distributed data system.

Jin teaches a server architecture wherein application can be run either in-process or out-of-process with the server program (see abstract). Jin teaches in-process node and out-of-process node are in the distributed data system (col. 6 lines 42-48).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify AAPA to include both in-process and out-of process nodes as in Jin. One would be motivated to do so to offer the flexibility to run either or both in-process and out-of-process applications (Jin, col. 6 lines 56-57).

Regarding claim 18, AAPA teaches the method as recited in claim 17, further comprising: the distributed data manager for the in-process node receiving a request for data from another node; the distributed data manager for the in-process node serializing the requested data; and the distributed data manager for the in-process node sending

the serialized data to the requesting node (page 4 lines 2-15).

Claim 19 does not recite or define any new limitation above claim 4 and therefore is rejected for similar reason.

Claim 20 does not recite or define any new limitation above claim 5 and therefore is rejected for similar reason.

Claim 21 does not recite or define any new limitation above claim 6 and therefore is rejected for similar reason.

Claim 22 does not recite or define any new limitation above claim 7 and therefore is rejected for similar reason.

Claim 23 does not recite or define any new limitation above claim 8 and therefore is rejected for similar reason.

Claim 24 does not recite or define any new limitation above claim 9 and therefore is rejected for similar reason.

Claim 28 does not recite or define any new limitation above claim 13 and therefore is rejected for similar reason.

Claim 29 does not recite or define any new limitation above claim 14 and therefore is rejected for similar reason.

Claim 30 does not recite or define any new limitation above claim 15 and therefore is rejected for similar reason.

Claim 31 does not teach or define any new limitation above claim 16 and therefore is rejected for similar reason.

Regarding claim 32, AAPA teaches a method, comprising:

an out-of-process client requesting data from a distributed data manager for an out-of-process node of a distributed data system (page 2 lines 20-21), wherein the out-of-process client and the distributed data manager for the out-of-process node execute in two distinct processes (page 1 lines 30-31);

if the requested data is present in a data store managed by the distributed data manager for the out-of-process node:

the distributed data manager for the out-of-process node returning the requested data to the out-of-process client as a serialized object (page 2 lines 20-23 and page 3 lines 1-3);

if the requested data is not present in the data store managed by the distributed data manager for the out-of-process node:

the distributed data manager for the out-of-process node retrieving the requested data in a serialized format from another node of the distributed data system (page 2 lines 23-24 and page 3 lines 1-3); and

the distributed data manager for the out-of-process node returning the requested data in a serialized format to the out-of-process client (page 2 lines 27-28);

an in-process client requesting data from a distributed data manager for an in-process node of the distributed data system (page 4 lines 4-5), wherein the in-process client and the distributed data manager for the in-process node execute within the same process on the in-process node (Fig. 2, page 3 lines 15-16); and

the in-process client receiving the requested data in de-serialized format page 4 lines 14-18).

AAPA does not explicitly teach in-process node and out-of-process node are in the distributed data system.

Jin teaches a server architecture wherein application can be run either in-process or out-of-process with the server program (see abstract). Jin teaches in-process node and out-of-process node are in the distributed data system (col. 6 lines 42-48).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of AAPA to include both in-process and out-of process nodes as in Jin. One would be motivated to do so to provide higher performance at the risk of crashing the system and afford high reliability (**Jin**, col. 4 lines 55-62).

Claim 33 does not recite or define any new limitation above claim 10 and therefore is rejected for similar reason.

2. Claims 10-12 and 25-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Admitted Prior Art (AAPA) in view of Jin, and further in view of Crites et al. (Crites) (US 6,097,380).

Regarding claim 10, AAPA teaches the system as recited in claim 7, wherein said out-of-process client is configured to send serialized data to the distributed data manager for the out-of-process node to store data to the distributed data manager for the out-of-process node (page 2 lines 14-16 and page 2 line 30-page 3 line3),

AAPA-Jin does not explicitly teach the distributed data manager for the out-of-process node is configured to store the data in its serialized format.

Crites teaches a network system wherein one or more servers and a plurality of available media streams is included (see abstract). Crites teaches the distributed data manager for the out-of-process node is configured to store the data in its serialized format (col. 2 lines 36-42 and col. 5 line19-20).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify AAPA-Jin to store the data in its serialized format as in Crites. One would be motivated to do so to enable application programs to render continuous media streams of different types and from different sources without being aware of the details

of the media streams (Crites, col. 1lines 40-43).

Claim 11 does not recite or define any new limitation above claim 10 and therefore is rejected for similar reasons.

Claim 12 does not recite or define any new limitation above claim 10 and therefore is rejected for similar reasons.

Claim 25 does not recite or define any new limitation above claim 10 and therefore is rejected for similar reasons.

Claim 26 does not recite or define any new limitation above claim 10 and therefore is rejected for similar reasons.

Claim 27 does not recite or define any new limitation above claim 10 and therefore is rejected for similar reasons.

(10) Response to Argument

In the argument, appellant argued in substances that

The rejection is improper because the art relied upon by the examiner explicitly teaches away from a distributed data system comprising a plurality of nodes including at least one in-process node and at least one out-of-process

node, as recited in claim 1.

(A) The AAPA reference relied upon by the Examiner in rejecting claim 1 explicitly teaches away from the combination of limitation in claim 1.

As to point (**A**), Examiner respectfully submits that AAPA teaches in-process node and out-of-process node (page 1 paragraph [0004]-page 4 paragraph [0013]). Jin teaches a system that can run/configure both in-process and out-process applications/nodes (col. 6 lines 42-57). It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of AAPA to run both in-process and out-of process applications in the same system as taught by Jin. One would be motivated to do so to provide higher performance at the risk of crashing the system and afford high reliability (**Jin**, col. 4 lines 55-62). Therefore, examiner respectfully submits that the rejection is proper because the art relied upon by examiner does not teach way from a distributed data system comprising a plurality of nodes including at least one in-process node and at least one out-of-process node.

(**B**) Jin does not teach or suggest an in-process node of a distributed data system, where the in-process node comprises a client and a distributed data manager configured to execute within the same process.

As to point (**B**), Examiner asserted in Final Office Action dated March 21, 2006 that AAPA teaches an in-process node of a distributed data system, where the in-process node comprises a client and a distributed data manager configured to execute

within the same process *(i.e., in the in-process configuration data may be communicated between a distributed data manager and a client sharing the same process space, Fig. 2 page 3 paragraph [0009] and page 4 paragraph [0012])*. Thus, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

(C) It is improper to combine references where the references teach away from their combination.

As to point (**C**), Examiner asserts that AAPA teaches in-process node and out-of-process node (page 1 paragraph [0004]-page 4 paragraph [0013]). Jin teaches a system that can run/configure both in-process and out-process applications/nodes (col. 6 lines 42-57). It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of AAPA to run both in-process and out-of process applications in the same system as taught by Jin. One would be motivated to do so to provide higher performance at the risk of crashing the system and afford high reliability (**Jin**, col. 4 lines 55-62). Therefore, examiner respectfully submits that it is proper to combine references where the references do not teach away from their combination.

(D) The combination of AAPA and Jin fails to teach or suggest wherein all data store operations performed by the distributed manager in the in-process node store data in a non-serialized object format in a data store of the in-process node.

As to point (**D**), examiner respectfully submits that AAPA teaches all data store operations performed by the distributed data manager in the in-process node store data in a non-serialized object format in a data store of the in-process node (i.e., AAPA discloses in the in-process configuration data may be communicated between (data store of it corresponding) distributed data manager and a client *without* requirement for s*erialization/deserialization*, pages 3-4 paragraphs [0011]-[0012. One of ordinary skill in the art will readily recognize that all data store operations performed by the distributed manager in the in-process node store data in a non-serialized format since communication between a distributed data manager and client without the additional communication requirement for serialization/deserialization).

(**E**) Examiner's interpretation, storing data in serialized format, is incorrect.

As to point (**E**), Examiner has given a broadest reasonable interpretation of "data in serialized format" as "data stream" in view of Appellant's specification *(i.e., **serialization** may including generating object data sequentially so that it may be transmitted **as data stream**, page 3 lines 7-8)*. Therefore, examiner respectfully submits that examiner's interpretation is correct.

(**F**) AAPA fails to teach the claimed limitation wherein the out-of-process client is configured to send serialized data to the distributed manager for the out-of-process node to store data.

As to point (**F**), Examiner respectfully submits that AAPA does teach the claimed limitation wherein the out-of-process client is configured to send serialized data to the distributed manager for the out-of-process node to store data (*i.e., client may send data to distributed manager. Distributed data manager may store data in its data store, wherein data is serialized before transmission, transmitted and received in its serialized format, pages 2-3 paragraphs [0006]-[0007]).*

(**G**) It would not render the combination of limitation of claim 10 obvious. Thus the rejection of claim 10 is clearly unsupported by the cited prior.

As to point (**G**), Examiner respectfully submits that that AAPA teaches in-process node and out-of-process node (page 1 paragraph [0004]-page 4 paragraph [0013]). Jin teaches a system that can run/configure both in-process and out-process applications/nodes (col. 6 lines 42-57). Crites teaches store the data in its serialized format (i.e., store stream data, col. 2 lines 36-42 and col. 5 line19-20). It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of AAPA and Jin to store the data in its serialized format as in Crites. One would be motivated to do so to enable application programs to render continuous media streams of different types and from different sources without being aware of the details of the media streams (Crites, col. 1lines 40-43). Therefore, it would render the

combination of limitations of claim 10 obvious. Thus the rejection of claim 10 is clearly supported by the cited prior art.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Oanh Duong
November 20, 2006

Conferees:

Lynne H. Browne
Appeal Specialist, TQAS
Technology Center 2100

SALEH NAJJAR
SUPERVISORY PATENT EXAMINER